

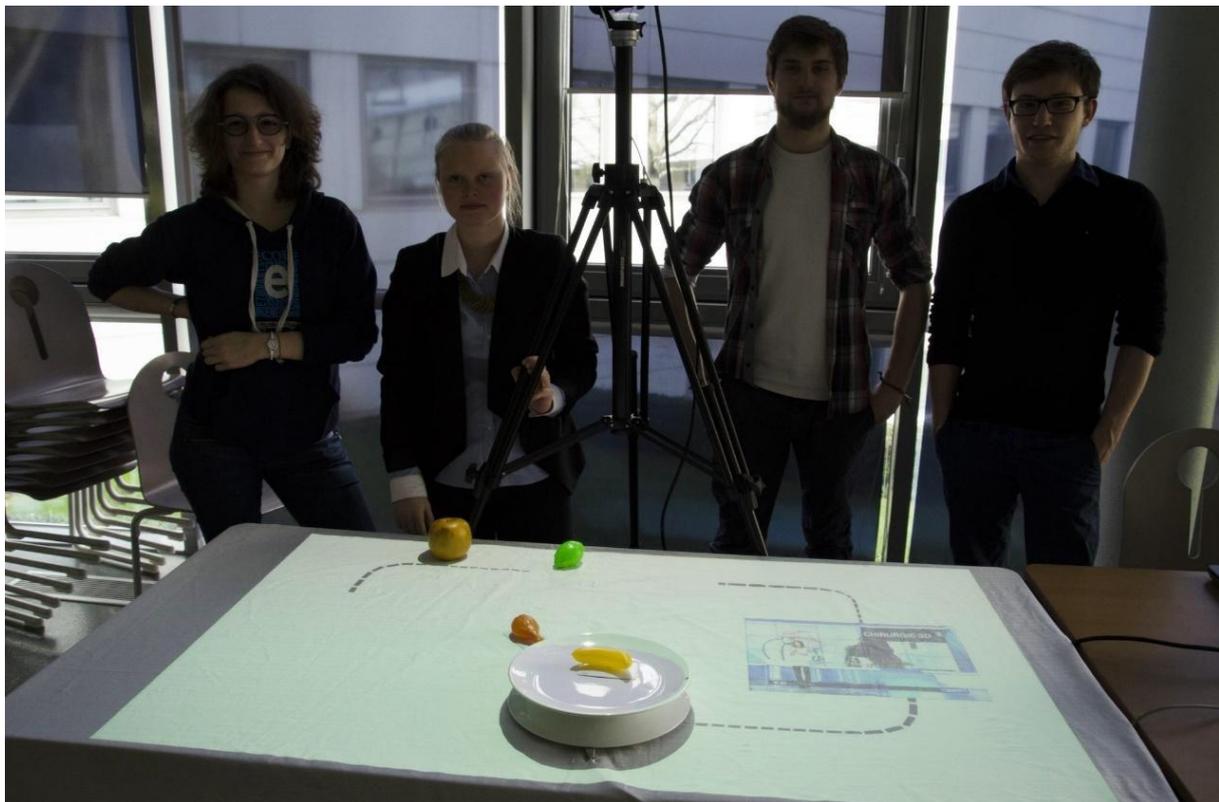
BOURDAIS Jeanne
CHICOISNE Killian
LEHMSTEDT Sébastien
MARGUERITE Marine



M^{me} LABBÉ
M. LE RENARD

CCSTI CREATION D'UNE TABLE NUMERIQUE

PST 4A – 2015/2016



CCSTI

Centre de Culture Scientifique,
Technique et Industrielle de LAVAL

Table des matières

1.	Introduction.....	3
2.	Remerciements	4
3.	Présentation de l'équipe	5
4.	Présentation du projet	6
a.	Etat de l'art.....	6
b.	Pourquoi ce projet?.....	7
c.	Présentation du projet	8
d.	Comment allons-nous travailler durant le projet.....	9
e.	Les étapes et jalons du projet	10
5.	Réalisation du projet	11
a.	Détection de l'aliment.....	11
•	Côté application Unity3D.....	11
•	Côté carte Arduino	12
b.	La machine à états.....	13
c.	Chargement des animations	15
d.	Affichage du contenu	16
e.	Création et chargement des chemins	17
f.	Problèmes rencontrés	18
6.	Conclusion	20
7.	Bibliographie.....	21
a.	Articles	21
b.	Sites Web.....	21
8.	Annexes	22
a.	Tutoriel pour la communication entre Unity3D et l'Arduino	22
•	Paramètres Unity pour utiliser le Port Série.....	22
•	Pour importer la librairie dans Arduino :.....	23
b.	Tutoriel pour ajouter un aliment ou modifier du contenu	24
•	Ajout d'un aliment dans la base de données	25
•	Modification d'un contenu.....	26

1. Introduction

Ce projet est réalisé en collaboration avec le CCSTI (Centre de Culture Scientifique, Technique et Industrielle) de Laval, dirigée par Mme LABBE. Il s'agit de développer un dispositif de réalité augmentée qui se veut intuitif et ludique ayant pour objectif premier de sensibiliser les utilisateurs à ce qui se trouve dans leurs assiettes. L'utilisateur pose un aliment dans l'assiette et toutes sortes d'informations s'afficheront sur la table à propos de l'aliment. Celles-ci pourront être sous forme de contenu visuel ou textuel.

Ce projet a pour finalité d'être présenté lors d'une exposition au Musée des Sciences de Laval qui aura lieu à partir de novembre 2016. L'exposition portera sur "L'innovation en agroalimentaire" et répondra à quatre thématiques :

"L'agroalimentaire c'est quoi ? Comment ça marche ?"

"Manger hier, manger demain"

"Innover c'est quoi, innover pourquoi ?"

"Bien manger pourquoi comment ?"

Ce projet peut convenir à un public large et varié, désireux de s'informer, ou seulement curieux, sur l'origine des aliments proposés dans les grandes surfaces et chez les commerçants, pour savoir un peu mieux ce qu'ils peuvent s'attendre à trouver dans leurs assiettes.

2.Remerciements

Nous tenons tout d'abord à remercier Mme LABBE pour nous avoir soumis ce projet intéressant qui regroupe aussi divers domaines d'étude qui nous ont motivés à réaliser le projet, comme l'utilisation de la technologie RFID ainsi que l'utilisation de logiciel de modélisation 3D pour obtenir de la réalité augmentée. Nous la remercions aussi pour toutes les informations qu'elle a pu nous fournir pour le bon déroulement du projet et de nous avoir permis d'accéder au musée afin de mieux voir les contraintes à prendre en compte pour que notre projet soit viable pour être exposé.

Nous voulons aussi remercier Mr LE RENARD pour son aide dans le projet, sa présence et son soutien dans notre travail, ainsi que pour les nombreuses fois où il nous a aiguillés sur les produits nécessaires et utiles à notre projet et sur la manière de le mettre en œuvre.

3.Présentation de l'équipe

Notre groupe est composé de quatre personnes :

- BOURDAIS Jeanne
- CHICOISNE Killian
- LEHMSTEDT Sébastien
- MARGUERITE Marine

Nous pouvons aussi inclure dans l'équipe, Mme LABBE pour avoir pensé ce projet et qui nous encadre dans sa réalisation, ainsi que Mr LE RENARD, notre suiveur de projet.

Nous sommes quatre étudiants de 4e année, passionnés par les nouvelles technologies dans de nombreux domaines que nous découvrons à l'ESIEA.

Nous nous sommes tous dirigés vers les systèmes d'information (SI), ce que nous pensons être un atout dans ce projet qui ne contient que très peu d'électronique, et qui requiert surtout des connaissances dans la programmation et une touche de créativité.

4. Présentation du projet

a. Etat de l'art

De nombreuses applications de réalité augmentée existent aujourd'hui. On les retrouve d'ailleurs sous différentes formes, que ce soit sur smartphone ou sur ordinateur. Mais peu existent en rapport avec le thème de la nourriture. En effet, des chercheurs japonais se sont penchés sur la question et ont d'ores et déjà développé une application à l'aide d'un casque composé d'une caméra et d'un logiciel, permettant de changer la perception d'un biscuit afin de mesurer l'impact de la quantité de nourriture sur notre faim. Ils ont remarqué que plus la quantité est petite dans notre assiette, plus nous avons faim, et inversement.

Sur le même principe, Tetra Pak a développé une application mobile, permettant à l'utilisateur (le public ciblé étant les enfants de 6 à 12 ans) de voir s'animer en 3D des personnages sur la brique de lait, suivant 15 scénarios différents traitant de la protection, de la gestion des forêts, du tri, du recyclage ou encore de la biodiversité. Malheureusement ces applications ne fonctionnent que pour un type d'aliment, celui prédéfini, l'utilisateur ne peut changer d'aliment pour avoir d'autres informations.

b. Pourquoi ce projet?

Dans un premier temps, nous avons choisi ce projet, car nous trouvons intéressant d'appliquer nos connaissances ainsi que celles que l'on va acquérir sur un projet qui a pour but de sensibiliser les gens à ce qu'ils mangent. En effet nous pensons que de nos jours, beaucoup de consommateurs sont préoccupés par ce qu'il y a dans leurs assiettes. Cette préoccupation survient à cause de certains scandales comme celui de la viande de cheval (Findus), ou encore des problèmes d'hygiène importants dans des entreprises de l'industrie agroalimentaire, en plus des maladies qui touchent les animaux ou encore des pesticides dans les fruits et légumes.

Nous pensons que le consommateur est déboussolé (comme nous le sommes) et ne sait pas si ce qu'il mange est vraiment bon pour sa santé. En somme, on peut dire qu'il ne fait plus vraiment confiance à ce qu'il mange. Avec notre projet, nous voulons ainsi montrer aux consommateurs la réalité du système agroalimentaire en lui proposant de visionner le parcours étape par étape, complété du nombre de kilomètres parcourus par le produit et ses valeurs nutritives ainsi que toutes autres informations utiles qui nous seront fournies par M^{me} LABBE.

De plus ce projet regroupe des domaines qui nous intéressaient comme la technologie de communication par RFID ou encore la découverte et l'utilisation de logiciels de modélisation comme Unity3D, qui s'avèrent être des outils utiles à maîtriser de nos jours.

c. Présentation du projet

L'objectif de ce projet est la réalisation d'une table numérique pour sensibiliser l'utilisateur au contenu de son assiette (apports nutritionnels, provenance, conditions d'élevage ou de culture).

Concrètement ce projet se présente sous la forme d'une table, avec une assiette dans laquelle des aliments factices seront déposés, ainsi que d'un vidéoprojecteur au-dessus de la table pour l'affichage des informations relatives à l'aliment déposé dans l'assiette.

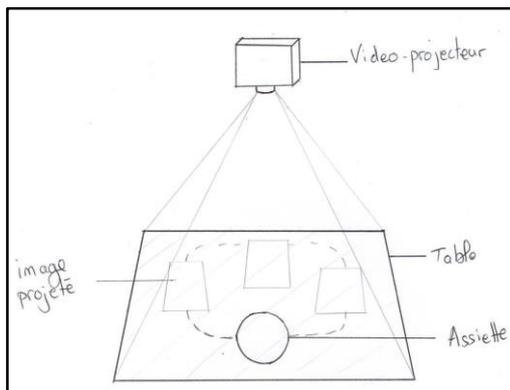


FIGURE 2. DESSIN REPRESENTANT LE PROJET

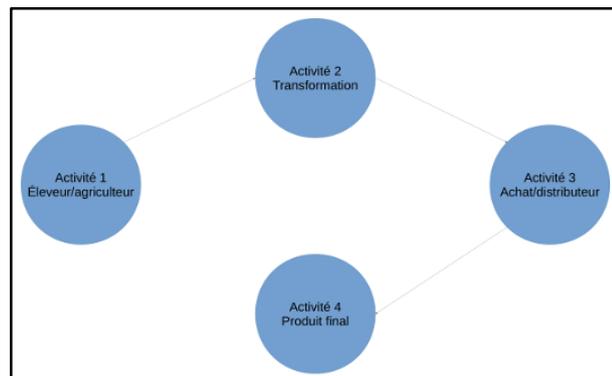


FIGURE 1. SCHEMA DU PLACEMENT DES ANIMATIONS

Afin d'avoir une idée précise du projet, faisons une étude pas à pas de son processus de fonctionnement :

- 1) L'utilisateur dépose un des aliments factices dans l'assiette.
- 2) Lecture de la puce RFID contenue dans l'aliment par le lecteur RFID qui envoie les informations à la carte Arduino.
- 3) Traitement des données par le programme Arduino.
- 4) Envoi des données par le port série à l'ordinateur.
- 5) Sélection dans un fichier XML du contenu à afficher en fonction du code RFID.
- 6) Génération de l'animation dans Unity3D avec le contenu correspondant à l'aliment.
- 7) Envoi du flux vidéo généré sous Unity3D au vidéoprojecteur.

d. Comment allons-nous travailler durant le projet

À la vue de l'envergure du projet, nous avons décidé de nous diviser en deux sous-groupes afin de mieux nous organiser et d'avancer plus rapidement.

Sur le projet, un premier groupe s'occupe des calculs préliminaires, comme les dimensions de la table et la hauteur du vidéoprojecteur, pour un affichage correct sur la table, ainsi que de la modélisation de la table sous Unity3D afin d'obtenir notre table en virtuel et "filmer" ce qui sera projeté sur notre vraie table. S'ajoute à cela la création des animations pour notre scène, et l'insertion du contenu fourni par M^{me} LABBE.

Le second groupe s'occupe de la récupération des informations d'une puce RFID par un lecteur RFID (shield Arduino) afin de savoir à quel produit le code correspond, ainsi que de la programmation de la carte Arduino pour traiter l'information contenue dans la puce pour enfin la renvoyer de façon correcte par port série.

Nous faisons également des points réguliers sur le projet et son avancée ainsi que sur les réalisations de chacun d'entre nous par un groupe Facebook spécialement mis en place pour ce projet. De plus, nous partageons la totalité du projet tenu à jour sur le drive de Google, ce qui nous permet d'être plus efficaces.

e. Les étapes et jalons du projet

Ce projet va être ponctué d'étapes et de jalons qui sont considérés comme une avancée dans notre travail. Notre groupe étant séparé en deux sous-groupes, nous avons défini des étapes pour chacun d'eux.

Partie RFID :

- Étape 1: Être en mesure de récupérer les informations de la puce RFID.
- Étape 2: Effectuer correctement le traitement des informations.
- Étape 2: Envoyer de manière fonctionnelle des informations par le port série à Unity3D.

Partie Unity3D :

- Étape 1: Calculer à quelle hauteur le vidéo projecteur sera positionné pour obtenir la taille de l'image et ainsi connaître les dimensions de la table.
- Étape 2: Réaliser un modèle virtuel de la table avec les bonnes dimensions sous Unity3D.
- Étape 3: Créer les animations.

La dernière étape sera sûrement commune où il s'agira de réaliser l'architecture des dossiers sur l'ordinateur pour un fonctionnement correct ainsi que le remplissage du fichier .XML qui contiendra toutes les informations utiles à notre projet.

5. Réalisation du projet

a. Détection de l'aliment

Pour réaliser la détection de l'aliment, nous utilisons la technologie RFID. Chaque aliment factice contient une puce RFID (sous forme de clous), et sous l'assiette est placée l'antenne RFID (module Adafruit) soudée à une carte Arduino Uno. Celle-ci est reliée à l'ordinateur où se trouve l'application.



FIGURE 3. CARTE ARDUINO ET MODULE RFID

Pour cette partie, nous avons donc dû travailler d'une part sur la carte Arduino, et d'une autre part sur Unity3D. Nous allons tout d'abord expliquer l'algorithme du code Unity3D puis nous détaillerons celui de la carte Arduino.

- Côté application Unity3D

En premier lieu, notre programme sur Unity3D cherche sur quel port est branchée la carte Arduino. Pour cela on envoie le message « ccsti » sur chaque port ouvert trouvé. Si l'on obtient une réponse, cela veut dire qu'elle se trouve au port concerné. Après initialisation d'autres paramètres, on envoie un message « START » à la carte Arduino pour qu'elle commence à récupérer les codes RFID.

L'algorithme qui suit se trouve dans un thread, et est donc exécuté continuellement en tâche de fond. Dans ce thread, l'application attend donc de recevoir des informations de la part de la carte Arduino, via le port série. Lorsque des octets sont disponibles à la lecture dans le port série, on vérifie s'il s'agit du code « FFFFFFFF » qui signifie que plus aucune puce RFID n'est posée dans l'assiette. On vérifie ensuite si l'identifiant RFID reçu existe dans le fichier XML et correspond donc à un aliment.

Dans tous les cas, on appelle ensuite la gestion des états (de la machine à états, qui sera expliquée ci-dessous) pour qu'elle réagisse en fonction de l'information récoltée.

Pour finir, lorsque notre application est coupée, elle envoie le message « STOP » à l'Arduino pour lui demander d'arrêter l'envoi des codes RFID.

- Côté carte Arduino

Au niveau de la carte Arduino, le programme vérifie le port série continuellement. S'il y lit quelque chose, trois messages l'intéressent. Tout d'abord le message « ccsti », auquel il répond, indiquant ainsi à notre application sur Unity3D que la carte est sur ce port. Ensuite, s'il reçoit le message « START », cela veut dire que notre algorithme de récupération du code RFID et de son envoi peut s'exécuter en continu. Finalement, si la carte Arduino reçoit le message « STOP », le programme ne fera plus rien d'autre que d'écouter le port série en attendant un nouveau message « START ».

L'algorithme exécuté en continu consiste à récupérer l'identifiant RFID que transmet lui module RFID et à le traiter.

Si une puce RFID est laissée posée sur l'antenne, celle-ci communiquera continuellement l'identifiant de la puce. Notre programme garde donc en mémoire le dernier identifiant reçu. Si la même puce est toujours détectée, le code sera le même, on ne le renverra donc pas. Si l'identifiant est différent, cela veut dire qu'une autre puce est posée : on envoie à l'application Unity3D via le port série le nouvel identifiant détecté. On peut également savoir si aucun aliment n'est posé dans l'assiette : on envoie alors le code « FFFFFFFF » pour prévenir l'application.

Finalement au niveau technique, notre module RFID utilise le protocole MiFare et la fréquence 13.56Mhz. Notre programme gère des identifiants RFID de seulement quatre octets.

b. La machine à états

Afin de gérer de manière robuste tous les cas d'utilisation possibles de notre matériel, nous avons sur les conseils de M Le Renard développé une machine à états. En effet le programme développé sur Unity3D répète le code de manière séquentielle dans sa boucle principale, Update. L'utilisateur pouvant potentiellement faire ce qu'il souhaite avec les aliments factices, il est primordial que le système ne présente aucun comportement indéterminé en cas de retrait, de remplacement soudain ou d'ajout d'aliments dans l'assiette.

La machine présente trois états principaux : ATTENTE, CHARGEMENT et AFFICHAGE.

Lorsque la machine est "au repos", c'est à dire lorsqu'aucune carte n'est détectée par le module RFID, elle est en état ATTENTE.

Dans la boucle Update du programme, lorsqu'une carte est détectée, sa validité est vérifiée (son code RFID doit être présent dans le fichier de configuration "conf.xml"). Le cas échéant l'appel à notre fonction de machine à états, nommée "Gestion_Des_Etats", est effectué. Dans ce cas, la machine passe en état CHARGEMENT.

De par la fonction du projet, qui présente le cycle de vie d'un aliment choisi par l'utilisateur parmi plusieurs, le système ne doit avoir à gérer qu'une seule carte et un seul aliment à la fois. C'est pourquoi notre fonction de machine à états vérifie si pendant un certain intervalle de temps (trois secondes), elle ne détecte qu'une seule carte, c'est à dire si le code RFID identifié par le module ne varie pas. Si au contraire, elle en détecte plusieurs, elle renvoie un message d'erreur, réinitialise le nombre de cartes détectées à zéro, et annule le timer représentant l'intervalle de détection. Dans le où la carte est retirée, la carte Arduino envoie à notre application le code RFID "FFFFFFF", et, quel que soit l'état dans lequel se trouve la machine, celle-ci revient en état ATTENTE.

Au bout des trois secondes de détection de la même carte, l'aliment est validé par le programme et la machine à états passe en état AFFICHAGE. A ce moment, dans cet état, huit sous-états correspondent au cycle de vie de l'aliment : ASSIETTEACHAT, ACHAT, ACHATATRANSF, TRANSFORMATEUR, TRANSFAPROD, PRODUCTEUR, ASSIETTE et FINAFFICHAGE. Le découpage de cet état permet de gérer les allocations de mémoire (textures, textes, vidéos) de manière stable pour chaque étape du trajet d'un aliment.

L'intérêt d'une machine à états est sa flexibilité. En effet, quelque que soit l'état et/ou le sous-état actuel de la machine, le retrait ou l'addition d'une carte entraînera toujours les mêmes conséquences sur le système. Cette absence de comportement indéterminé est le critère qui nous permet de garantir la robustesse de notre application.

c. Chargement des animations

Nous avons décidé de n'utiliser que des vidéos pour les trois animations concernant respectivement la production, la transformation et la distribution. Chaque aliment possède donc trois vidéos ainsi que des informations textuelles. Nous avons donc créé un fichier XML contenant chaque aliment, et pour chacun de ceux-ci les liens vers les vidéos et les informations.

Toutes les vidéos sont chargées en mémoire au démarrage de notre application. Pour mener à bien ce chargement, dans notre programme une liste de nos aliments a été faite. En effet, pour accéder à chaque vidéo à partir de notre application, il est nécessaire que nous soyons en mesure de les retrouver facilement dans la mémoire, ce qui a motivé notre choix de recourir aux listes. Notre liste contient les aliments qui sont dans notre projet, et chaque aliment contient une liste de trois vidéos, un nom, une valeur énergétique (en lipides, glucides et protides), ainsi que le nombre de kilomètres parcourus entre chaque étape et la quantité de CO2 rejetée associée. Chaque aliment est également lié à une liste de codes RFID correspondant aux codes des puces RFID que nos aliments sont susceptibles de contenir.

Pour charger le contenu pendant le lancement de notre application, la compréhension du fonctionnement des threads et coroutines de Unity3D fut nécessaire. En effet, une coroutine commence seulement à la fin de la fonction de notre boucle de jeu, en arrière-plan. Il n'y a donc aucun moyen de savoir quand elle commence, cela dépend de son ordre dans la pile d'appels de Unity3D. Dans notre projet, nous avons une vingtaine de vidéos à charger, soit vingt coroutines à lancer, gérer, et dont les résultats (les vidéos chargées en mémoire) sont à remplir dans notre liste. Nous devons attendre que le chargement de toutes les vidéos d'un aliment soit terminé pour commencer à charger les prochaines vidéos de l'aliment suivant. Cette étape fut assez difficile à mettre en œuvre en raison de sa complexité. En effet, si jamais le chargement d'une vidéo ne démarre pas au bon moment, l'organisation de notre liste devient obsolète et toutes nos données sont mélangées.

d. Affichage du contenu

Une fois que le chargement du contenu du projet fonctionne correctement, l'affichage sur la table lors de la détection d'un transpondeur RFID est relativement aisé et lié à notre machine à état. Les listes de vidéos étant pré-remplies, une fois qu'une vidéo est jouée, le programme sait directement laquelle est la suivante.

Pour un affichage structuré sur la table, nous avons effectué de nombreux calculs pour optimiser l'espace disponible et obtenir un espace dédié aux vidéos aussi grand que possible (Figure 4). Les espaces disponibles entre les vidéos ont pu servir pour afficher nos chemins entre les vidéos. Pour les informations textuelles comme le kilométrage ou l'empreinte carbone, ou encore les logos de l'école et du CCSTI, l'espace blanc est utilisé.

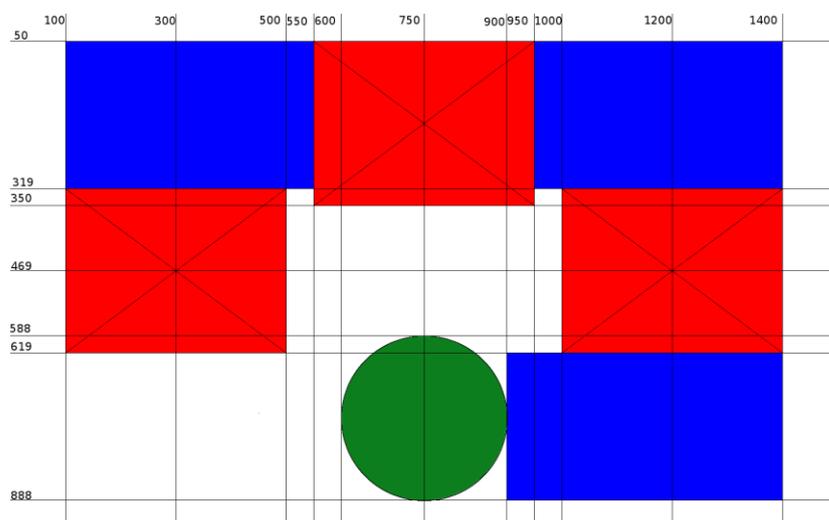


FIGURE 4. EMPLACEMENT DES DIFFERENTES ACTIVITES (ROND VERT POUR L'ASSIETTE, RECTANGLES ROUGES POUR LES VIDEOS ET RECTANGLES BLEUS POUR LES CHEMINS)

Enfin, nous avons pensé à l'utilisateur, et réalisé à l'emplacement de l'assiette une animation lui permettant de suivre la détection de son aliment. L'assiette devient verte lorsque l'aliment est détecté. En cas d'erreur, si l'utilisateur a posé deux objets dans l'assiette, celle-ci clignote en rouge et un message d'erreur est affiché à l'utilisateur afin de lui renseigner l'usage correct de notre dispositif.

e. Création et chargement des chemins

La dernière partie qui a été faite est la création et le chargement des chemins entre les vidéos. Pour rendre notre projet plus graphique et attrayant, les chemins qui s'afficheront entre chaque vidéo sont animés. Nous avons donc créé avec Photoshop trois chemins, le premier de notre assiette à la première vidéo concernant la distribution, puis un second de cette vidéo à celle sur la transformation et le troisième chemin de cette dernière à la vidéo sur la production. Cela permet de mettre en avant la continuité du cycle de vie des aliments. Les chemins sont faits d'une quinzaine d'images, affichées les unes après les autres afin de former l'animation d'un trajet. De plus une image représentant l'aliment suivra le chemin tracé.

f. Problèmes rencontrés

Pour la partie RFID, le premier cas d'erreur géré a été la détection du port série sur lequel la carte Arduino est branchée. En effet ce n'est pas le même d'un ordinateur à un autre.

Le second problème, correspondant au cas où plus d'un aliment est posé dans l'assiette, est géré dans Unity3D. La carte Arduino reçoit les deux identifiants RFID et les envoie continuellement sur le port série. Notre application attend trois secondes depuis le dernier identifiant reçu. Si un nouvel identifiant est détecté, on remet à zéro le compteur.

Finalement, pour cette partie, le dernier problème fut au moment de la commande de clous RFID. Nous avons travaillé toute l'année avec une carte RFID (fournie avec le modèle), mais il nous fallait un autre type de transpondeur pour nos aliments. Mr GAGEOT nous ayant conseillé les clous RFID, nous avons fait attention à la taille en octets de l'identifiant RFID, mais nous n'avons pas pensé à regarder le protocole utilisé ni la fréquence correspondante. Nous en avons donc recommandé.

Pour charger des vidéos dans Unity3D, la version professionnelle de Unity3D était normalement requise pour effectuer cette opération. Finalement, à partir de l'été 2015 l'utilisation de la vidéo dans Unity3D est passée dans la partie libre du logiciel, et des recherches approfondies nous ont finalement permis d'afficher de la vidéo dans ce logiciel.

Ensuite, une autre problématique est apparue : toutes les vidéos doivent être chargées au préalable avant d'être lues. Pour un fichier vidéo de taille modeste, le temps de chargement est négligeable, mais pour une vidéo durant une ou deux minutes, le temps devient un paramètre à prendre en compte. Ce fait nous a amené à mettre en œuvre un processus de chargement des contenus au démarrage de notre application Unity3D.

Une autre difficulté fut la prise en main des threads et des coroutines de Unity3D qui sont relativement peu documentées sur Internet, malgré leur intérêt certain dans la parallélisation des tâches.

Finalement, un dernier problème est le côté pratique : pour une personne non technique, il est très difficile de pouvoir ajouter un aliment ou modifier les informations relatives à un autre. Il faudrait concevoir une interface simple permettant de remplir des champs prédéfinis, et qui générerait la partie xml correspondante dans notre fichier.

6. Conclusion

Pour conclure, nous pouvons dire que ce projet est la réalisation d'un travail d'équipe, et nous avons pu réaliser un projet professionnel qui tenait à cœur à nos superviseurs, Mme LABBE et M. LE RENARD. Malgré quelques problèmes, nous avons pu utiliser les moyens à notre disposition pour les résoudre et aller de l'avant.

Ce projet nous a initiés à la tenue et au respect d'un cahier des charges, avec un professionnel comptant sur notre travail pour le succès de son exposition.

Le projet nous a aussi fait découvrir la réalité virtuelle et nous a fait prendre conscience de la variété des domaines dans laquelle celle-ci s'inscrit. La réalité augmentée n'est pas seulement ludique, elle peut revêtir des aspects pédagogiques et formateurs, comme nous avons pu le constater en l'appliquant nous-mêmes dans ces champs.

7. Bibliographie

a. Articles

Britvic et Orlait animent leurs packs par réalité augmentée, K.E. | 15 Septembre 2015

<http://www.processalimentaire.com/Flash-IAA/Britvic-et-Orlait-animent-leurs-packs-par-realite-augmentee-27051>

La réalité augmentée au service des régimes alimentaires du futur, Florian LOPEZ | 6 novembre 2012

<http://www.actinnovation.com/innovation-technologie/realite-augmentee-regimes-alimentaires-futur-5068.html>

b. Sites Web

Site d'Adafruit (module RFID)

<https://learn.adafruit.com/adafruit-pn532-rfid-nfc>

Site Arduino

<https://www.arduino.cc/>

Site Unity3D

<http://unity3d.com/>

8. Annexes

a. Tutoriel pour la communication entre Unity3D et l'Arduino

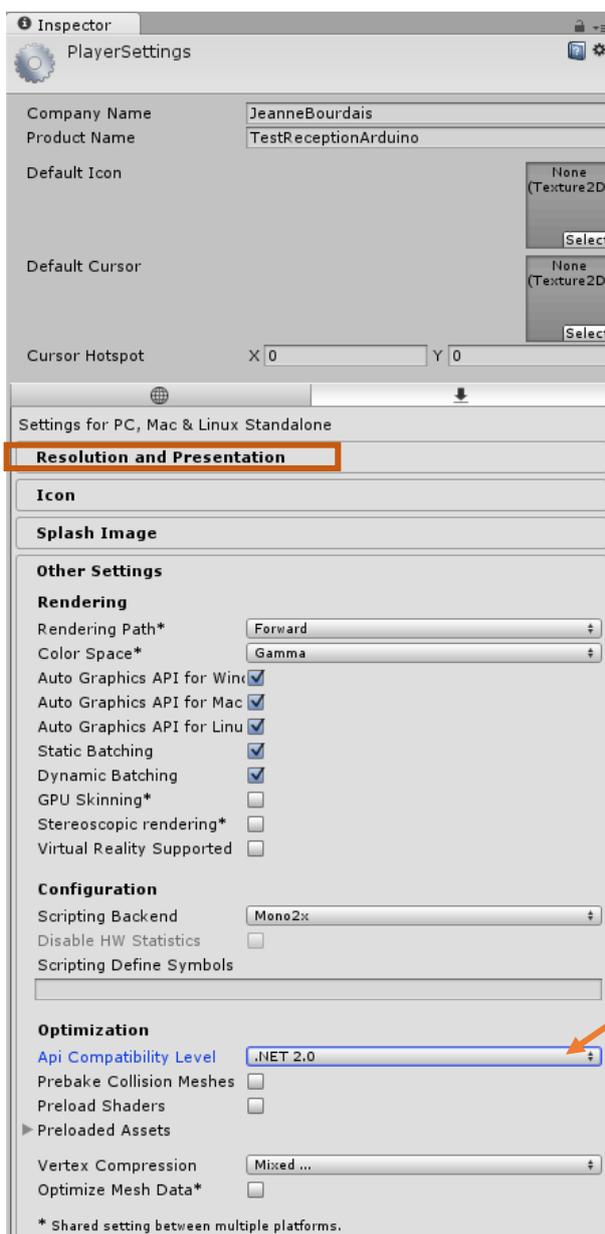
- Paramètres Unity pour utiliser le Port Série

1 – Un seul logiciel peut écouter le port (ne pas lancer le moniteur d'Arduino si on veut pouvoir accéder au Port via Unity)

2 – Modification nécessaire d'un paramètre : Edit → Project Settings → Player

Dans l'inspector, aller dans l'onglet « Settings for PC » et dans la partie « Optimization » modifier l'Api Compatibility Level en « .NET 2.0 » et non « .NET 2.0 Subset » (CF image)

3 – Recharger le script.



- Pour importer la librairie dans Arduino :

1 – Télécharger le fichier Adafruit_PN532.ZIP, disponible sur <https://github.com/adafruit/Adafruit-PN532/>

2 – Ouvrir Arduino, aller dans Croquis → Inclure une bibliothèque → Ajouter la bibliothèque .ZIP ...

3 – Sélectionner le fichier .ZIP que l'on vient de télécharger.

b. Tutoriel pour ajouter un aliment ou modifier du contenu

```

1  <?xml version="1.0" encoding="utf-8"?>
2
3  <AlimentsCollection>
4
5      <Aliments>
6
7          <!-- DEBUT DU BLOC -->
8          <!-- CODE DE L'ALIMENT -->
9          <!-- poulet -->
10         <Aliment code="0">
11             <RFIDs>
12                 <RFID>
13                     <!-- CODE RFID DU CLOU CORRESPONDANT -->
14                     <code_RFID>724B49A5</code_RFID>
15                 </RFID>
16             </RFIDs>
17
18             <!-- NOMBRE DE PROTIDES (%) DANS L'ALIMENT -->
19             <Protides>2</Protides>
20             <!-- NOMBRE DE LIPIDES (%) DANS L'ALIMENT -->
21             <Lipides>3</Lipides>
22             <!-- NOMBRE DE GLUCIDES (%) DANS L'ALIMENT -->
23             <Glucides>4</Glucides>
24
25             <Cadres>
26                 <Cadre activite="vendeur">
27
28                     <!-- CHEMIN DE L'IMAGE CORRESPONDANT AU VENDEUR -->
29                     <Image>\ressources\poulet.png</Image>
30                     <!-- CHEMIN DE LA VIDEO CORRESPONDANT AU VENDEUR -->
31                     <Video>\ressources\TF1cervo0.ogv</Video>
32                     <!-- INFORMATION SUR LE TRAJET DE L'ASSIETTE AU VENDEUR -->
33                     <Trajet distance="15" empreinte_carbone="7"/>
34                 </Cadre>
35
36                 <Cadre activite="transformateur">
37
38                     <!-- CHEMIN DE L'IMAGE CORRESPONDANT AU TRANSFORMATEUR -->
39                     <Image>\ressources\poulet2.png</Image>
40                     <!-- CHEMIN DE LA VIDEO CORRESPONDANT AU TRANSFORMATEUR -->
41                     <Video>\ressources\TF1cervo1.ogv</Video>
42                     <!-- INFORMATION SUR LE TRAJET DU VENDEUR AU TRANSFORMATEUR -->
43                     <Trajet distance="20" empreinte_carbone="5"/>
44                 </Cadre>
45
46                 <Cadre activite="producteur">
47                     <!-- CHEMIN DE L'IMAGE CORRESPONDANT AU PRODUCTEUR -->
48                     <Image>\ressources\poulet3.png</Image>
49                     <!-- CHEMIN DE LA VIDEO CORRESPONDANT AU PRODUCTEUR -->
50                     <Video>\ressources\TF1cervo2.ogv</Video>
51                     <!-- INFORMATION SUR LE TRAJET DU TRANSFORMATEUR AU PRODUCTEUR -->
52                     <Trajet distance="50" empreinte_carbone="10"/>
53                 </Cadre>
54             </Cadres>
55         </Aliment>
56         <!-- FIN DU BLOC -->
57
58         <!-- FIN DU FICHER -->
59     </Aliments>
60 </AlimentsCollection>

```

- Ajout d'un aliment dans la base de données

1. Mettre le contenu (images et vidéos) relatif à l'aliment dans le dossier « ressources » situé à la racine de notre projet.
2. Toujours à la racine de notre projet, ouvrir le fichier conf.xml (avec notepad++ ça marche très bien).
3. Dans le fichier conf.xml, le premier bloc aliment est commenté.
4. Vous pouvez le copier/coller à la fin du fichier .xml, avant le commentaire <!--FIN DU FICHER -->.

Il faudra ensuite modifier tous les champs commentés, voici la liste :

```
<Aliment code="0">
```

Ce code correspond au nombre d'aliment créé - 1. Il faut donc regarder quel est le code du dernier aliment dans le fichier xml et l'incrémenter pour notre nouvel aliment.

```
<code_RFID>724B49A5</code_RFID>
```

Ce code correspond à l'identifiant RFID du clou.

```
<Protides>2</Protides>
```

```
<Lipides>3</Lipides>
```

```
<Glucides>4</Glucides>
```

Ces trois champs correspondent au nombre de grammes de chaque nutriment dans l'aliment.

```
<Image>\ressources\poulet.png</Image>
```

```
<Video>\ressources\TF1cervo0.ogv</Video>
```

```
<Trajet distance="15" empreinte_carbone="7"/>
```

Ces trois champs sont à remplir trois fois : une fois pour les informations sur la vente, une seconde fois pour la transformation et une dernière fois pour la production.

Les deux premiers champs correspondent à l'image et à la vidéo et que vous avez ajouté précédemment dans le dossier « ressources ».

Notez qu'un petit commentaire pour préciser quel aliment ces informations concernent serait le bienvenu ! Pour faire un commentaire : <!-- commentaire -->

- [Modification d'un contenu](#)

Pour modifier un contenu, s'il s'agit d'une image ou d'une vidéo, elle est à placer dans le dossier « ressources » situé à la racine de notre projet.

Il faut ensuite ouvrir le fichier conf.xml, également situé à la racine du projet.

Dans ce fichier, retrouvez le bloc aliment correspondant à celui que vous voulez modifier. Il suffit ensuite de modifier la ligne voulue. Pour plus de facilité, regardez les commentaires du premier bloc aliment.